The program written is using LinkedList to read and print polynomials whose term are given by the user. Also, to performs functions like appending and deleting a term to any given polynomial. The functions to add or multiply any two given polynomials are implemented. The program contains functions to,

- reading user's input of the terms of the polynomial – polyin()
- display the polynomial – polyout()
- add any two given polynomials – polyadd()
- multiply any two given polynomials – polymul()
- function to store value of polynomials in above operations – insertterm()
- add a new term to the existing polynomial – appendterm()
- delete any term in the existing polynomial – deleteterm()

An essential characteristic of the polynomial is that each term in the polynomial expression consists of two parts: one is the coefficient and the other is the exponent. So, a node structure having coefficients, exponents and each node pointing to the address of the next term is created.

Addition Approach:

- If power of 1st polynomial is greater than 2nd, then store 1st as it is and move its pointer.
- If power of 2nd polynomial is greater than 1st, then store 2nd as it is and move its pointer.
- If power of both polynomial numbers is same then, add their coefficients.
- Store the added value in a new linked list i.e., p3.
- We will move the pointers of first and second polynomial to get its next term.

Multiplication Approach:

- We will multiply the coefficient of the 2nd polynomial with each term's coefficient of the 1st polynomial.
- We will add the coefficients of elements having the same exponential power in resultant polynomial.
- Store the multiplied value in a new linked list i.e., p4.

Appending Approach:

- Ask the user for the input towards which polynomial among the two should the new term be added.
- If the exponential of the new term is equal to exponential of any existing terms in the given polynomial, then we add their coefficients.
- Store the new term added to the existing polynomial in the existing linked list.

Deletion Approach:

- To delete a term from any of the two polynomials, required is the user input to the position of the term to be deleted. If the node to be deleted is the first node, then we point head to the next node i.e., second node.
- In other positions, we find the previous node of the node to be deleted and change the next of the previous node to the next node of the deleted node.

Sorting Approach:

- If the exponential of j is greater than exponential of temp, then we swap the exponential of temp in t1, t2 to exponential of j in t1, t2 respectively. We repeat this for each term in the polynomial.